Exploiting Memory Content Similarity to Improve Memory Performance in Large-Scale Distributed Systems

Scott Levy, *University of New Mexico* (slevy@cs.unm.edu) / **Kurt B. Ferreira**, *Sandia National Laboratories* / **Patrick G. Bridges**, *University of New Mexico*

THE UNIVERSITY of NEW MEXICO

Motivation

As we consider building the next generation of extreme-scale systems, many of the biggest challenges are related to memory characteristics. In particular, overcoming challenges related to resilience and memory bandwidth will require innovative strategies for improving the performance of main memory.

DRAM Failures

- one of the most frequently observed sources of node failure in large-scale distributed systems [2]
- failure rates increase proportionally to the number of processors
- as systems grow, traditional approaches to fault tolerance (e.g., coordinated checkpoint/restart) may no longer be sufficient [3]
- deploying low voltage memory chips may exacerbate this problem [4]

Silent Data Corruption

- ECC cannot catch all memory errors
- an incorrect result may be the only indication of a failure [8]

Memory Bandwidth

- limited memory bandwidth will restrict our ability to fully exploit the increasingly powerful multicore processors that will compose future systems [1]
- clock rates have plateaued, growth in total computational power has been achieved by increasing the number of cores per processor
- number of cores is growing more rapidly than memory access
- effective memory bandwidth per core is decreasing over time.

Application Suite

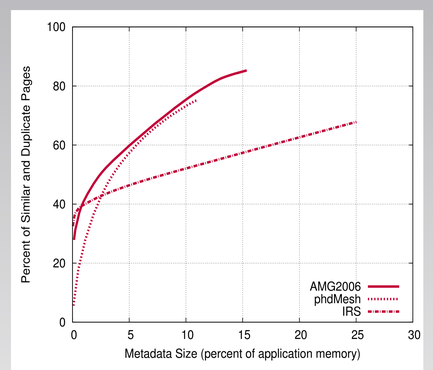
ASC Sequoia Marquee Performance Codes	AMG	Parallel algebraic multigrid solver
	IRS	Implicit Radiation Solver; radiation transport
DOE Production Applications	CTH	Multi-material, large deformation, strong shock wave, solid mechanics code
	LAMMPS	Molecular dynamics simulator
Mantevo Mini-Applications	HPCCG	Mimics finite element generation, assembly and solution for an unstructured grid problem
	phdMesh	Mimics the contact search applications in an explicit finite element application
Miscellaneous Applications	SAMRAI	Enables application of structured adaptive mesh refinement to large-scale multi-physics problems
	Sweep3D	Neutron transport problem

Modification Behavior

Application	Changed 1+ Times	Changed 1 Time	Changed 2 Times	Changed 3 Times	Changed 4+ Times
AMG2006	20.8 %	9.9 %	5.4 %	1.7 %	3.8 %
CTH	38.9 %	6.9 %	3.5 %	13.7 %	14.8 %
IRS	32.8 %	18.3 %	0.2 %	0.0 %	14.3 %
LAMMPS	37.6 %	0.5 %	0.6 %	0.5 %	36.0 %
SAMRAI	79.5 %	13.6 %	7.7 %	32.0 %	26.3 %
HPCCG	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
phdMesh	21.6 %	6.2 %	1.9 %	0.5 %	13.0 %
Sweep3D	4.1 %	1.7 %	0.7 %	0.0 %	1.8 %

For the majority of the applications that we considered, a substantial majority of pages that are *ever* classified as similar or duplicate are modified either once or not at all. This suggests that the overhead of maintaining metadata may be manageable.

Metadata Size



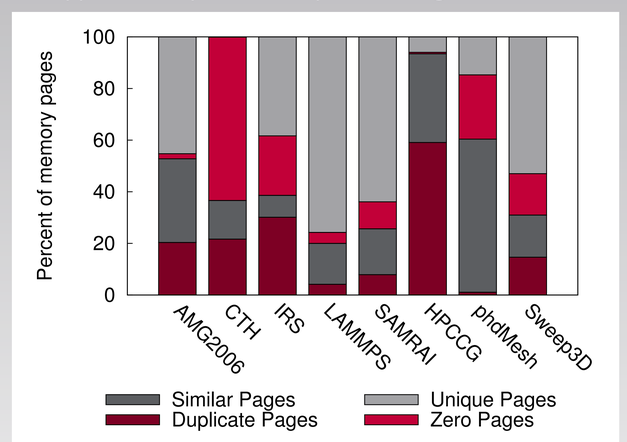
Exploiting content similarity requires that we maintain some metadata about similar pages. There is a tradeoff to be made between the size of the metadata and the number of similar and duplicate pages. As shown in this figure, increasing the patch size threshold yields more similar pages, but requires more metadata.

Resilience

Memory content similarity may enable us to withstand memory errors without requiring a restart. Content similarity can also be exploited to protect against silent data corruption.

Prevalence of Similarity

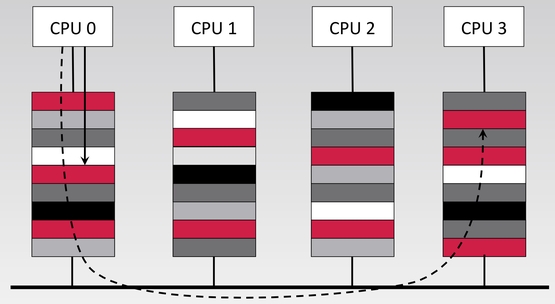
As shown in the figure below, less than 50% of the memory of many applications is comprised of unique pages. This suggests that our approach may be able to provide a significant benefit.



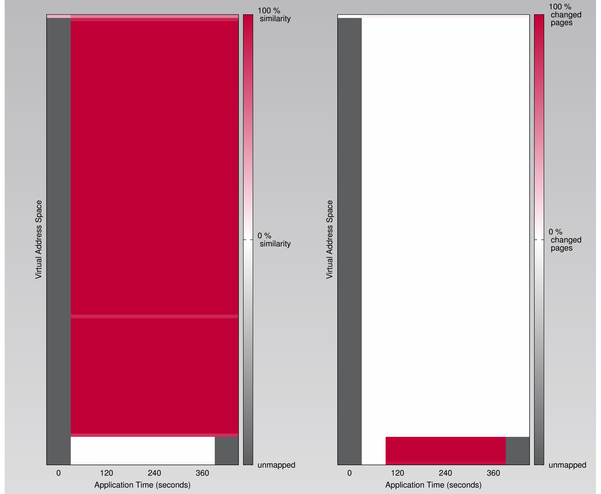
- **ZERO**: all-zero pages
- **DUPLICATE**: non-zero pages whose contents match at least one other page
- **SIMILAR**: those pages: (a) that are neither duplicate nor zero; and (b) can be paired with a another page such that the difference between the two can be captured in a `cx_bsdiff` patch smaller than 1024 bytes
- **UNIQUE**: all other pages

NUMA Memory

Exploiting content similarity may allow us to replace accesses to pages in slow, remote memory (DASHED) with accesses to pages in fast, local memory (SOLID)



Application Structure



SIMILARITY MODIFICATION

HPCCG

Given the picture that emerges from these figures, we speculate that because HPCCG is a conjugate gradient solver, the low end of the virtual address space (top of the figure) contains the sparse matrix that is provided as input and is never modified and the high virtual addresses contain the solution vector that is refined on each iteration. We intend to in-depth investigation of the sources of similarity in several applications

References

[1] V. Chandra and R. Aitken, "Impact of technology and voltage scaling on the soft error susceptibility in nanoscale CMOS," in *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS'08. IEEE International Symposium on*. IEEE, 2008, pp. 114–122.
[2] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN2006)*, Jun. 2006. [Online]. Available: http://www.pdl.cmu.edu/PDL-FTP/stray/dsn06 abs.html
[3] K. Ferreira, R. Riesen, J. Stearley, J. H. L. III, R. Oldfield, K. Pedretti, P. Bridges, D. Arnold, and R. Brightwell, "Evaluating the viability of process replication reliability for exascale systems," in *Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis, (SC'11)*, Nov 2011.
[4] C. Lu and D. Reed, "Assessing fault sensitivity in MPI applications," in *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2004, p. 37.
[5] B. Rogers, A. Krishna, G. Bell, K. Vu, X. Jiang, and Y. Solihin, "Scaling the bandwidth wall: challenges in and avenues for CMP scaling," in *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3. ACM, 2009, pp. 371–382.